

1 **CLAIMS**

2
3 1. A software architecture for a distributed computing system
4 comprising:

5 an application configured to handle requests submitted by applications
6 executing on remote devices over a network; and

7 an application program interface to present functions used by the
8 applications to access network and computing resources of the distributed
9 computing system, wherein the application program interface comprises a set of
10 base classes and types that are used in substantially all applications executing on
11 the remote devices submitting requests.
12

13 2. A software architecture as recited in claim 1, wherein the set of base
14 types comprises:

15 an AsyncCallback delegate supplied to an application, wherein the
16 AsyncCallback delegate references a callback method to be called when a
17 corresponding asynchronous operation is completed; and

18 an IAsyncResult interface that enables determination of the status of an
19 asynchronous operation.
20

21 3. A software architecture as recited in claim 2, wherein the
22 IAsyncResult interface includes:

23 an AsyncState property that returns the object that was provided as the last
24 parameter as part of a Begin call corresponding to the asynchronous operation;
25

1 an AsyncWaitHandle property that returns a WaitHandle that can be used to
2 allow the application to wait for a call to be completed without needing to poll;

3 a CompletedSynchronously property that is set to true if the Begin call
4 corresponding to the asynchronous operation completed synchronously; and

5 an IsCompleted property that is set to true after processing of the
6 asynchronous operation is completed.

7
8 **4.** A software architecture as recited in claim 1, wherein the set of types
9 support an event model including an event delegate that connects an event with a
10 handler of the event, the set of base classes and types further comprising:

11 one or more classes that hold event data; and

12 one or more delegates that identify a method to provide a response to an
13 event.

14
15 **5.** A software architecture as recited in claim 1, wherein the application
16 program interface further comprises a collections namespace that includes a
17 plurality of classes and interfaces for in-memory data storage and manipulation.

18
19 **6.** A software architecture as recited in claim 5, wherein the collections
20 namespace includes, as at least part of the plurality of types:

21 a first set of types including commonly used collection classes;

22 a second set of types including interfaces to define a formal contract
23 between developers creating new collections and developers consuming
24 collections; and

25 a third set of types that support creating strongly typed collections.

1
2 7. A software architecture as recited in claim 1, wherein the application
3 program interface further comprises a globalization namespace that includes a
4 plurality of classes that define culture-related information, wherein the plurality of
5 classes include a first set of types representing information about a user's culture
6 and a second set of types representing information about a user's region.

7
8 8. A software architecture as recited in claim 1, wherein the application
9 program interface further comprises a net namespace that includes a plurality of
10 classes that enables use of network resources without details of one or more
11 protocols used to access the network resources.

12
13 9. A software architecture as recited in claim 1, wherein the application
14 program interface further comprises a security namespace that includes a plurality
15 of classes and interfaces that make available an underlying structure of a security
16 system including one or more cryptographic services, code access security and
17 role based security infrastructure.

18
19 10. A software architecture as recited in claim 1, wherein the
20 application program interface further comprises a service process namespace that
21 includes a plurality of classes that allow installation and running of services.

1 11. A software architecture as recited in claim 1, wherein the
2 application program interface further comprises a serialization namespace that
3 includes a plurality of classes that enable serializing and deserializing of instance
4 data.

5
6 12. A software architecture as recited in claim 1, wherein the
7 application program interface further comprises a diagnostics namespace that
8 includes a plurality of classes that enable debugging of applications, trace code
9 execution, reading event logs, writing event logs, and monitoring system
10 performance.

11
12 13. A software architecture as recited in claim 1, wherein the
13 application program interface further comprises a messaging namespace that
14 includes a plurality of classes that enable connecting to message queues on the
15 network, sending messages to message queues, receiving messages from message
16 queues, and peeking at messages from message queues.

17
18 14. An application program interface, embodied on one or more
19 computer readable media, comprising:

20 an AsyncCallback delegate supplied to an application, wherein the
21 AsyncCallback delegate references a callback method to be called when a
22 corresponding asynchronous operation is completed; and

23 an IAsyncResult interface that enables determination of the status of an
24 asynchronous operation.
25

1 **15.** An application program interface as recited in claim 14, wherein the
2 IAsyncResult interface includes:

3 an AsyncState property that returns the object that was provided as the last
4 parameter as part of a Begin call corresponding to the asynchronous operation;

5 an AsyncWaitHandle property that returns a WaitHandle that can be used to
6 allow the application to wait for a call to be completed without needing to poll;

7 a CompletedSynchronously property that is set to true if the Begin call
8 corresponding to the asynchronous operation completed synchronously; and

9 an IsCompleted property that is set to true after processing of the
10 asynchronous operation is completed.

11
12 **16.** A distributed computer software architecture, comprising:

13 one or more applications configured to be executed on one or more
14 computing devices, the applications handling requests submitted from remote
15 computing devices;

16 a networking platform to support the one or more applications; and

17 an application programming interface to interface the one or more
18 applications with the networking platform, wherein the application program
19 interface comprises a set of types that are used in each of the one or more
20 applications.

1 **17.** A distributed computer software architecture as recited in claim 16,
2 wherein the set of base classes and types comprises:

3 an AsyncCallback delegate supplied to an application, wherein the
4 AsyncCallback delegate references a callback method to be called when a
5 corresponding asynchronous operation is completed; and

6 an IAsyncResult interface that enables determination of the status of an
7 asynchronous operation.

8
9 **18.** A method comprising:

10 receiving one or more application program interface (API) calls from one or
11 more remote devices over a network, wherein the one or more calls are to one or
12 more functions that include a set of base classes and types that are used in
13 substantially all applications executing on the one or more remote devices; and

14 performing the function requested in each of the one or more calls.

15
16 **19.** A method as recited in claim 18, wherein the set of base classes and
17 types comprises:

18 an AsyncCallback delegate supplied to an application, wherein the
19 AsyncCallback delegate references a callback method to be called when a
20 corresponding asynchronous operation is completed; and

21 an IAsyncResult interface that enables determination of the status of an
22 asynchronous operation.

1 **20.** A method as recited in claim 18, wherein the set of base classes and
2 types support an event model including an event delegate that connects an event
3 with a handler of the event, the set of base classes and types further comprising:

4 one or more classes that hold event data; and

5 one or more delegates that identify a method to provide a response to an
6 event.

7
8 **21.** A method as recited in claim 18, wherein the application program
9 interface further comprises a collections namespace that includes a plurality of
10 classes and interfaces for in-memory data storage and manipulation.

11
12 **22.** A method as recited in claim 18, wherein the application program
13 interface further comprises a globalization namespace that includes a plurality of
14 classes that define culture-related information, wherein the plurality of classes
15 include a first set of types representing information about a user's culture and a
16 second set of types representing information about a user's region.

17
18 **23.** A method as recited in claim 18, wherein the application program
19 interface further comprises a net namespace that includes a plurality of classes that
20 enables use of Internet resources without details of one or more protocols used to
21 access the Internet resources.

22
23 **24.** A method comprising:
24 calling, to one or more remote devices over a network, one or more
25 functions via an application program interface (API) that make available a set of

1 base classes and types that are used in substantially all applications calling the one
2 or more functions;

3 receiving, from the one or more remote devices, a response to the calling.
4

5 **25.** A method as recited in claim 24, wherein the set of base classes and
6 types comprises:

7 an AsyncCallback delegate supplied to an application, wherein the
8 AsyncCallback delegate references a callback method to be called when a
9 corresponding asynchronous operation is completed; and

10 an IAsyncResult interface that enables determination of the status of an
11 asynchronous operation.
12

13 **26.** A method as recited in claim 24, wherein the set of types support an
14 event model including an event delegate that connects an event with a handler of
15 the event, the set of base classes and types further comprising:

16 one or more classes that hold event data; and

17 one or more delegates that identify a method to provide a response to an
18 event.
19

20 **27.** A method as recited in claim 24, wherein the application program
21 interface further comprises a collections namespace that includes a plurality of
22 classes and interfaces for in-memory data storage and manipulation.
23
24
25

1 **28.** A method as recited in claim 24, wherein the application program
2 interface further comprises a globalization namespace that includes a plurality of
3 classes that define culture-related information, wherein the plurality of classes
4 include a first set of types representing information about a user's culture and a
5 second set of types representing information about a user's region.

6
7 **29.** A method as recited in claim 24, wherein the application program
8 interface further comprises a net namespace that includes a plurality of classes that
9 enables use of Internet resources without details of one or more protocols used to
10 access the Internet resources.